Greg Gutierrez

Basic Structures of the C# Programming Language (MO 1.3)

Language Map for C#

<table>
<tr><td>Variable Declaration
Is this language strongly typed or dynamically typed? Provide an example of how variables are declared in this language.</td><td>C+ just like Java is a strongly typed language, meaning you have to declare the data type of your variables before you use them.
Example:
string name=”Sam”;
int num= 15;</td></tr>
<tr><td>Data Types
List all of the data types (and ranges) supported by this language.</td><td>
<table>
<tr><th>Data Types</th><th>Size</th><th>Description</th></tr>
<tr><td>int</td><td>4 bytes</td><td>Stores whole numbers from -2,147,483,648 to 2,147,483,647</td></tr>
<tr><td>long</td><td>8 bytes</td><td>Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807</td></tr>
<tr><td>float</td><td>4 bytes</td><td>Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits</td></tr>
<tr><td>double</td><td>8 bytes</td><td>Stores fractional numbers. Sufficient for storing 15 decimal digits</td></tr>
<tr><td>bool</td><td>1 bit</td><td>Stores true or false values</td></tr>
<tr><td>char</td><td>2 bytes</td><td>Stores a single character/letter, surrounded by single quotes</td></tr>
<tr><td>string</td><td>2 bytes per character</td><td>Stores a sequence of characters, surrounded by double quotes</td></tr>
</table>
</td></tr>
<tr><td>Selection Structures</td><td>C# supports the usual logical conditions from mathematics:</td></tr>
</table>

Provide examples of all selection structures supported by this language (if, if else, etc.)	Less than: a < b
	Less than or equal to: a <= b
	Greater than: a > b
	Greater than or equal to: a >= b
	Equal to a == b
	Not Equal to: a != b
	
	C# has the following conditional statements:
	
	Use **if** to specify a block of code to be executed, if a specified condition is true
	Use **else** to specify a block of code to be executed, if the same condition is false
	Use **else if** to specify a new condition to test, if the first condition is false
	Use **switch** to specify many alternative blocks of code to be executed
	
	Example for **if** statement :
	
	int x = 23;
	int y = 10;
	if (x > y)
	{
	Console.WriteLine("x is greater than y");
	}
	
	Output: x is greater than y
	Example for **else** statement:
	int clock= 22;
	if (time >=12)
	{
	Console.WriteLine("The day is beginning.");
	}
	else
	{
	Console.WriteLine("The day is almost coming to an end.");
	}

	Output: The day is almost coming to an end.
	
	Example for **else if** statement:
	
	int time = 22;
	if (time < 10)
	{
	  Console.WriteLine("Good morning.");
	}
	else if (time < 20)
	{
	  Console.WriteLine("Good day.");
	}
	else
	{
	  Console.WriteLine("Good evening.");
	}
	Output: Good evening.
	Example of a **switch** statement:
	
	int day = 5;
	switch (day)
	{
	  case 1:
	    Console.WriteLine("Monday");
	    break;
	  case 2:
	    Console.WriteLine("Tuesday");
	    break;
	  case 3:
	    Console.WriteLine("Wednesday");
	    break;
	  case 4:
	    Console.WriteLine("Thursday");
	    break;
	  case 5:
	    Console.WriteLine("Friday");
	    break;
	  case 6:

	Console.WriteLine("Saturday"); break; case 7: Console.WriteLine("Sunday"); break; } Output: Friday

Repetition Structures
Provide examples of all repetition structures supported by this language (loops, etc.)
	The **while** loop loops through a block of code as long as a specified condition is True:
	int i = 0;
	while (i < 5)
	{
	Console.WriteLine(i);
	i++;
	}
	
	Output:
	0
	1
	2
	3
	4
	
	The **do/while** loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.
	
	int i = 0;
	do
	{
	Console.WriteLine(i);
	i++;
	}
	while (i < 5);
	
	Output:
	0

	1
	2
	3
	4
	
	When you know exactly how many times you want to loop through a block of code, use the **for** loop instead of a while loop:
	
	for (int i = 0; i < 5; i++)
	{
	  Console.WriteLine(i);
	}
	
	Output:
	0
	1
	2
	3
	4
	
	foreach loop is used to loop through elements in an array
	
	string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
	foreach (string i in cars)
	{
	  Console.WriteLine(i);
	}
	
	Output:
	Volvo
	BMW
	Ford
	Mazda

Arrays *If this language supports arrays, provide an example of creating an array with a primitive data type (e.g. float, int, etc.)*	Yes, C# supports creating arrays. Example below: int[] num = {1, 2, 3, 4};
Data Structures *If this language provides a standard set of data structures, provide a list of the data structures and their Big-Oh complexity.*	Below is a picture that highlights the data structures used in C# and the algorithm efficiencies

Collection	Ordering	Contiguous Storage?	Direct Access?	Lookup Efficiency	Manipulate Efficiency	Notes
Dictionary	Unordered	Yes	Via Key	Key: O(1)	O(1)	Best for high performance lookups.
SortedDictionary	Sorted	No	Via Key	Key: O(log n)	O(log n)	Compromise of Dictionary speed and ordering, uses binary search tree.
SortedList	Sorted	Yes	Via Key	Key: O(log n)	O(n)	Very similar to SortedDictionary, except tree is implemented in an array, so has faster lookup on preloaded data, but slower loads.
List	User has precise control over element ordering	Yes	Via Index	Index: O(1) Value: O(n)	O(n)	Best for smaller lists where direct access required and no sorting.
LinkedList	User has precise control over element ordering	No	No	Value: O(n)	O(1)	Best for lists where inserting/deleting in middle is common and no direct access required.
HashSet	Unordered	Yes	Via Key	Key: O(1)	O(1)	Unique unordered collection, like a Dictionary except key and value are same object.
SortedSet	Sorted	No	Via Key	Key: O(log n)	O(log n)	Unique sorted collection, like SortedDictionary except key and value are same object.
Stack	LIFO	Yes	Only Top	Top: O(1)	O(1)*	Essentially same as List<T> except only process as LIFO
Queue	FIFO	Yes	Only Front	Front: O(1)	O(1)	Essentially same as List<T> except only process as FIFO

<table>
<tr>
<td>Objects
If this language support object-orientation, provide an example of how to create a simple object with a default constructor.</td>
<td>Yes C# supports object oriented programming:

REMINDER:
A constructor is a special method that is used to initialize objects. The advantage of a constructor, is that it is called when an object of a class is created. It can be used to set initial values for fields:

EXAMPLE:

<pre>class Car
{
  public string model; // Create a field

  // Create a class constructor for the Car class
  public Car()
  {
    model = "Mustang"; // Set the initial value for model
  }

  static void Main(string[] args)
  {
    Car Ford = new Car(); // Create an object of the Car Class (this will call the constructor)
    Console.WriteLine(Ford.model); // Print the value of model
  }
}</pre>

Output: Mustang</td>
</tr>
<tr>
<td>Runtime Environment
What runtime environment does this language compile to? For example, Java compiles to the Java Virtual Machine. Do other languages also compile to this runtime?</td>
<td>.NET’s runtime environment is called the Common Language Runtime (CLR). What this means in practice is that an application written in a language such as C# will be able to run on any piece of hardware that supports .NET. So, if you’re a developer using C# to write the code for your app, you know it’ll be able to run on anything that supports .NET.
Other language like Visual Basics and F# are written on .Net framework and are compiled with the Common Language Runtime as well.</td>
</tr>
<tr>
<td>Libraries/Frameworks
What are the popular libraries or frameworks used by programmers for this language? List at least three (3).</td>
<td>Just a note:
In simple terms, C# is a programming language, whereas .NET is the framework on which the language is built. Microsoft created .NET (Network Enabled Technology), and .NET developers will use programming languages such as C#. In fact, .NET supports many programming languages, and defines the rules and associated libraries those languages will use.
Some popular libraries in the .Net framework are:
Newtonsoft</td>
</tr>
</table>

	Orchard BetterCMS
Domains *What industries or domains use this programming language? Provide specific examples of companies that use this language and what they use it for.*	Many companies use C#. Many of these companies, although in different industries use C# for similar reasons. Taken from https://careerkarma.com/blog/who-uses-c-sharp/ Example: **ServiceTita**n is a rapidly growing software technology platform for trading. The ServiceTitan software is designed for organizations offering commercial HVAC, electrical, plumbing, and other services. This company uses C# to aid in developing its Android and web-based applications. **Stack Overflow** is a top website that serves over 100 million people per month. It is incredibly useful, especially for those who are learning to code because it allows them to share their knowledge and build their careers. The site is written in C#. **The Wintrust financial corporation** is a financial services provider based in Wisconsin. Like most banks and financial institutions, Wintrust uses C# primarily for front end web development, also known as client-side development. **Microsoft** is one of the world's leading computer and software companies. It developed C# to cope with the growing demands of web applications back in the year 2000. As a result, the company uses this programming language for the development of web services, games, and applications.